

**Amendments to the Claims**

1. (currently amended) A method of translating computer program code from a first language representation into a second language representation, the method comprising:  
translating ~~the~~ translatable instructions of an input stream in a first language representation into an output stream in a second language representation;  
identifying an unresolvable translation error in the input stream; and  
placing at least one second language representation instruction in the output stream responsive to identifying the unresolvable translation error in the input stream;  
wherein the placed at least one second language representation instruction is at least one of either a handling instruction or an exception throwing instruction.
2. (original) The method of claim 1 wherein said placing comprises placing the at least one second language representation instruction in a location in the output stream where the unresolvable translation error in the input stream would have been placed in the output stream had the unresolvable translation error been a translatable instruction.
3. (original) The method of claim 1 wherein said placing comprises placing the at least one second language representation instruction in a location in the output stream where a method containing the unresolvable translation error in the input stream would have been placed in the output stream had the entire method been translatable.
4. (original) The method of claim 1 wherein said placing comprises placing the at least one second language representation instruction in a location in the output stream where a basic block containing the unresolvable translation error in the input stream would have been placed in the output stream had the entire basic block been translatable.
5. (original) A method of executing the output stream of claim 1, the method comprising:  
executing at least one translated instruction and at least one placed second language representation instruction.

6. (original) The method of claim 1, wherein said placing further comprises:  
directing the placement of the at least one second language representation instruction within the output stream based on a declarative textual indication contained in the input stream.
7. (original) The method of claim 1, wherein said placing comprises placing both:  
at least one exception throwing instruction; and  
at least one handling instruction.
8. (currently amended) ~~The~~A method of executing the output stream of claim 7, the method comprising:  
executing the placed handling instruction subsequent to executing the placed exception throwing instruction.
9. (original) The method of claim 6, wherein the declarative textual indication designates that the at least one second language representation instruction replace an unresolvable translation error.
10. (original) The method of claim 6, wherein the declarative textual indication designates that the at least one second language representation instruction replace a basic block containing an unresolvable translation error.
11. (original) The method of claim 6, wherein the declarative textual indication designates that the at least one second language representation instruction replace a method containing an unresolvable translation error.
12. (original) The method of claim 1, further comprising:  
determining from a declarative textual indication in the input stream which at least one second language representation instruction to place in the output stream.
13. (currently amended) The method of claim 1, further comprising:  
obtaining from a library of available at least one second language representation instructions, the at least one second language representation instruction placed in the output stream.

14. (original) The method of claim 1, further comprising:  
wherein the at least one of either a handling instruction or an exception throwing instruction, is an application programming interface instruction to a dynamically linkable library.

15. (currently amended) A method of translating computer program code from an input stream in a first language representation into an output stream in a second language representation, and the input stream may or may not be from a trusted source, the method comprising:

translating ~~the~~ translatable instructions of the input stream into the output stream;  
identifying suspected code and unresolvable translation errors in the input stream;  
placing at least one second language representation instruction in the output stream responsive to identifying the unresolvable translation error in the input stream;  
determining that the input stream is from a trusted source; and  
translating the suspected code in the input stream into the output stream.

16. (original) The method of claim 15, wherein a declarative textual indication in the input stream is used to determine whether the input stream is from a trusted source.

17. (currently amended) The method of claim 15, wherein the determining further comprises:

making a request to a server; and  
determining from ~~the~~ server response whether the input stream is from a trusted source.

18. (original) The method of claim 5, wherein the presently executing instruction is the at least one placed second language representation instruction, the method further comprising:

requesting by the presently executing at least one placed second language representation instruction, the request being made to a server for instructions external to the output stream.

19. (original) The method of claim 1, further comprising:  
determining a level to place the at least one second language representation instruction;  
wherein the determination of level to place the at least one second language representation instruction is made from among a set of available levels, the set of available levels including at least two distinct levels from a group of potential levels, the group of potential levels comprising a method level, an instruction level, a basic block level, and a program level.

20. (original) The method of claim 19, wherein a declarative textual indication indicates the level to place the at least one second language representation instruction.

21. (currently amended) A method of translating computer program code from an input stream in a first language representation into an output stream in a second language representation, the input stream comprising declarative textual indications, the method comprising:

translating ~~the~~ translatable instructions in the input stream into the output stream;

identifying a first language representation of a declarative textual indication in the input stream, the declarative textual indication indicating how to handle ~~at least one of either an~~ unresolvable translation error ~~or a suspected code~~ encountered in the input stream; and

translating the first language representation of the declarative textual indication in the input stream into the second language representation of the declarative textual indications in the output stream;

whereby the second language representations of the declarative textual indications are available to a next phase of translation, the next phase of translation able to use the second language representation of the declarative textual indication as a resource for determining how to handle an unresolvable translation error ~~and/or suspect code~~ encountered by the next phase of translation as the next phase translates the output stream into a third language representation.

22. (currently amended) A method of translating the output stream of claim 21 into a third language output stream, the method comprising:

translating ~~the~~ translatable instructions of the output stream into the third language output stream;

identifying an unresolvable translation error in the output stream;

determining an indicated third language representation instruction from a declarative textual indication in the output stream; and

placing the indicated third language representation instruction in the third language output stream responsive to identifying the unresolvable translation error in the output stream.

23. (currently amended) A method of translating the output stream of claim 21 into a third language output stream, the method comprising:

translating ~~the~~ translatable instructions of the output stream into the third language output stream;

identifying a suspected code in the output stream;  
determining from the declarative textual indication in the output stream whether the output stream is from a trusted source; and  
translating the suspect code into the third language output stream in response to determining that the output stream is from the trusted source.

24. (currently amended) A method of translating computer program code from an input stream in a first language into an output stream in a second language, the method of translating including multiple sub-units of translation, the method comprising:

translating ~~the~~ translatable portions of the input stream into the output stream;  
identifying a suspected code in the input stream, determining by a security sub-unit of translation based on a declarative textual indication in the input stream that the input stream is from a trusted source, and translating the suspected code into the output stream;  
identifying a first unresolvable translation error in a first method in the input stream, determining by a basic block identification sub-unit of translation, that the first unresolvable translation error can not be safely segregated within a determinable basic block within the first method, and invoking a method level exception throwing instruction insertion sub-unit of translation to insert an exception throwing instruction in the output stream, and invoking a handler data structure insertion sub-unit of translation to insert a handler data structure in the output stream; and  
identifying a second unresolvable translation error in a second method in the input stream, determining by the basic block identification sub-unit of translation, that the second unresolvable translation error can be safely segregated within a determinable basic block within the second method, and invoking a basic block level exception throwing instruction insertion sub-unit of translation to insert an exception throwing instruction in the output stream, and invoking the handler data structure insertion sub-unit of translation to insert a handler data structure in the output stream.

25. (currently amended) In a computer readable medium, instructions operational to translate a first language input stream into a second language output stream, the instructions comprising:

instruction(s) for translating ~~the~~ translatable portions of a first language input stream into a second language output stream;

instruction(s) identifying an unresolvable translation error in the first language input stream;  
and

instruction(s) placing an exception throwing instruction in the second language output stream in response to identifying the unresolvable translation error in the first language input stream.

26. (currently amended) In a computer readable medium, instructions operational to translate a first language input stream into a second language output stream, the instructions comprising:

instruction(s) translating ~~the~~ translatable portions of a first language input stream into a second language output stream;

instruction(s) identifying suspected code and unresolvable translation errors in the first language input stream;

instruction(s) placing an exception throwing instruction in the second language output stream in response to identifying the unresolvable translation error in the first language input stream;

instruction(s) determining the source of the first language input stream; and

instructions(s) translating the suspected code into the second language output stream in response to determining that the first language input stream is from a trusted source.

27. (currently amended) In a computer readable medium, instructions operational to translate a first language input stream into a second language output stream, the instructions comprising:

instruction(s) for translating ~~the~~ translatable portions of a first language input stream into a second language output stream;

instruction(s) identifying an unresolvable translation error in the first language input stream;  
and

instruction(s) placing a handling instruction in the second language output stream in response to identifying the unresolvable translation error in the first language input stream.

28. (original) In a computer system having a user input device, a first language input stream, an operating system, a translation system operating under control of the operating system, the system comprising:

a translation process operating under control of the operating system, and translating the first language input stream into a second language output stream subsequent to a user input on the user input device;

a translation sub-unit operational to perform some portion of the translation process;  
and

a translation sub-unit operational to translate an unresolvable translation error identified in the first language input stream into an inserted exception throwing instruction in a translated code of the second language output stream.

29. (currently amended) In a computer system having a user input device, a first language input stream, an operating system, a translation system operating under control of the operating system, the system comprising:

a translation process operating under control of the operating system, and  
translating the first language input stream into a second language output stream subsequent to a user input on the user input device;

a translation sub-unit operational to translate an unresolvable translation error encountered in the first language input stream into an inserted handling instruction in a translated code of the second language output stream ~~perform some portion of the translation process~~; and

a translation sub-unit operational to verify that a source of the first language input stream is a trusted source, and translating a suspected code encountered in the first language input stream into a translated code in the second language output stream upon verifying that the source of the first language input stream is a trusted source.

30. (original) In a computer system having a user input device, a first language input stream, an operating system, a translation system operating under control of the operating system, the system comprising:

a translation process operating under control of the operating system, and translating the first language input stream into a second language output stream subsequent to a user input on the user input device;

a translation sub-unit operational to perform some portion of the translation process;  
and

a translation sub-unit operational to translate an unresolvable translation error encountered in the first language input stream into an inserted handling instruction in a translated code of the second language output stream.

31. (currently amended) The ~~system~~computer readable medium of claim 25, wherein the first language input stream further comprises a declarative textual indication indicating where in the output stream to place the exception throwing instruction.

32. (currently amended) The ~~system~~computer readable medium of claim 27, wherein the first language input stream further comprises a declarative textual indication indicating where in the output stream to place the handling instruction.

33. (original) A computer readable medium comprising translated code produced by claim 1.

34. (original) A method of delaying identification of compile time errors until run time, the method comprising:

compiling a first file containing a first language representation of a computer program into a second file containing a second language representation of the computer program;

identifying an unresolvable instruction in the first file;

determining where the identified unresolvable instruction in the first file would be placed in the second file by the compiler if the identified unresolvable instruction were resolvable;

translating the identified unresolvable instruction into a second language representation of an exception throwing instruction; and

inserting the second language representation of the exception throwing instruction into the second file where the identified unresolvable instruction would have been placed in the second file by the compiler if the unresolvable instruction were resolvable.

35. (original) The method of claim 1, further comprising:  
placing in the output stream an unaltered copy of the identified unresolvable translation error.



36. (original) A method of executing the output stream of claim 35, the method comprising:

executing the placed one second language representation instruction, said executing invoking a translator on the unaltered copy of the identified unresolvable translation error from the output stream, thereby causing the invoked translator to attempt to translate the unaltered copy of the unresolvable translation error into a second language representation instruction, and if the unaltered copy of the identified unresolvable translation error is translatable into a second language representation instruction, then the second language representation instruction is executed.

37. (original) The method of claim 1 wherein said placing comprises placing the at least one second language representation instruction in a new basic block in the output stream.

38. (original) The method of claim 6, wherein the declarative textual indication designates that the at least one second language representation instruction should be inserted in a new basic block.